Quiz1:-

Design (n-bit) 8-to-1 Mux using 8-to-1 Muxs as a component, and by using the generate and generic statments, then simulate your design for n = 8-bit. Note the (n-bit) 8-to-1 Mux has 8 inputs, each one is n-bit in addition to the selector (S), and the output Q is also n-bit.

First simulate the 8-to-1 Mux for all different values of the selector input, and then simulate the (n-bit) 8-to-1 Mux design for different combinations values of the inputs to verify the functionality of your design.

Submit word or pdf file that include all VHDL codes, RTL schematic, and simulation.

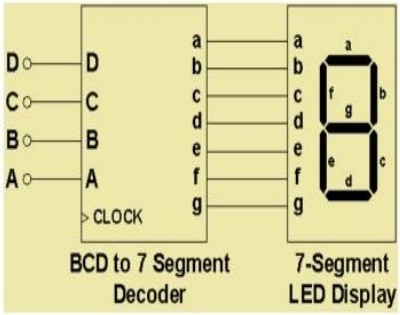
Test1:-

Q1- Design by using schematic only a logic unit that has four inputs, and one output, and can perform four different logic operations. The inputs are (a, b, s1, and s0), and the output is (q). The inputs (s1&s0) are used to select the required logic function. The circuit is working based on the following function table: [15-points]

en	s1 s0	Q (output function)
0	x x	'0'
1	0 0	Logic AND of a, b
1	0 1	Logic OR of a, b
1	1 0	Logic XOR of a, b
1	1 1	Logic XNOR of a, b

Simulate you design for all different input combinations of (s1, s0, a, b), when (en=0 and 1).

Q2- Design by using function the BCD to 7-segment decoder given in figure below: [15-points]



Simulate your design for all BCD values, with letter (E) for any other value.

Please submit one word or pdf file, include the schematic circuit and simulation of Q1, and RTL scematic, VHDL code (no screenshot), and simulation for Q2.

Assignment2:-

Modify the VHDL code for the RAM design given in the class today such that you can store the BCD codes for the 4-bit address. Write VHDL test bench for testing the design, in your VHDL test bench, first write the BCD codes in the RAM locations for all adresses starting from address "0000" up to address "1111", and then for reading five different memory locations.

Upload the simulation results in addition to the VHDL test bench code (no snapshot), all in one file.

Test2:-

Question (1):

A. Design by writing a complete VHDL code and by using components a parallel divider that is capable of dividing a positive 8-bit binary number by a positive 4-bit binary number to give a 4-bit quotient. Use the dividend register, divisor register, subtracor - comparator block, and control unit as components, the control unit will generate a signal "done", which will be used as indicator of finishing the division process, synthesis the code and show the RTL schematic. **[20 marks]**

B. Write a VHDL test bench file to simulate the circuit designed in (a) such that the following data fed into the circuit in sequence: **[10 marks]**

1. Divide 30 by 15.
2. Divide 15 by 3.
3. Divide 20 by 9.

Question (2):

A. Design 3*4 RAM, and test the RAM for writing and reading. **[10 marks]**

B. Design a parallel divider with memory element, use the 3*4 RAM with the parallel divider designed in Question (1), and any other additional components such that the first location of the RAM will be used to provide the divisor, and the second and third location will be used to store the quotient and remainder respectively. **[5 marks]**

C. Simulate the circuit that you have designed in (b). **[5 marks]**

Submit one word or pdf file that include all VHDL codes (do not Submit snapshot of the VHDL code), RTL schematic, and simulation.

Assignment3:-

Design a simple traffice light controller for two road with sensors that are working on the folowing timing sequences:

Ga+Rb (10-sec), Ya+Rb (1-sec), Ra+Rb (1-sec), Ra+Rb+Yb (1-sec), Ra+Gb (5-sec), Ra+Yb (1-sec), Ra+Rb (1-sec), Ra+Ya+Rb (1-sec), Ga+Rb (10-sec),...etc.

Use counter to reduce the number of states and check the required sensors in Ga+Rb, and Ra+Gb.

Write the VHDL test bench for testing your design, show the signal lightA and LightB in the simulation results.

Quiz2:-

1- Design 8-1 MUX with enable.

2- Design 8-bit binary up-down counter.

3- use the 8-1 MUX and 8-bit binary up-down counter as components, the 8-bit binary up-down counter will feed the data inputs of the MUX with clocks of different frequencies, use a proper connections between the counter and the MUX to show a clear simulation results, then simulate your design for all different combinations of the input variables with en=0 and en=1. The interface for the main design will be clk, en, selector as inputs, and q as output.

Final exam:-

Question 1 [30 points]

Design only the controller of a parallel binary divider that divide 4-bit dividend by a 2-bit divisor to obtain a 2-bit quotient and 2-bit remainder using a two address microprogramming; follow the following steps in your design:

1- Draw the block diagram for the two address microprogramming of the 4 by 2 divider controller, showing all inputs and outputs. **[3 points]**

2- Draw the two address microprogramming SM chart for the parallel binary divider control. **[3 points]**

3- Generate the microinstructions ROM table. **[4 points]**

4- Design the controller by writing a complete VHDL code and by using components according to the block diagram you have drawn in (1). **[10 points]**

5- Simulate your design for all input combinations, and show all the state transition in your simulation. **[10 points]**

Question2 [20 points]

Modify the design in question (1) to be implemented using single address microprogramming, follow the same steps in your design.

[Please submit the VHDL code (no snapshot), RTL, and simulation results as word or pdf file only through blackboard, any other file format will be ignored]